Delta Programming Task Documentation

Requirements:

- Allow the user to enter a station (destination or origin) to search flights. Display the results in a table.
- Provide an auto-suggest feature for station.
- Provide two RESTful endpoints supporting the functionality listed in steps 1 and 2.

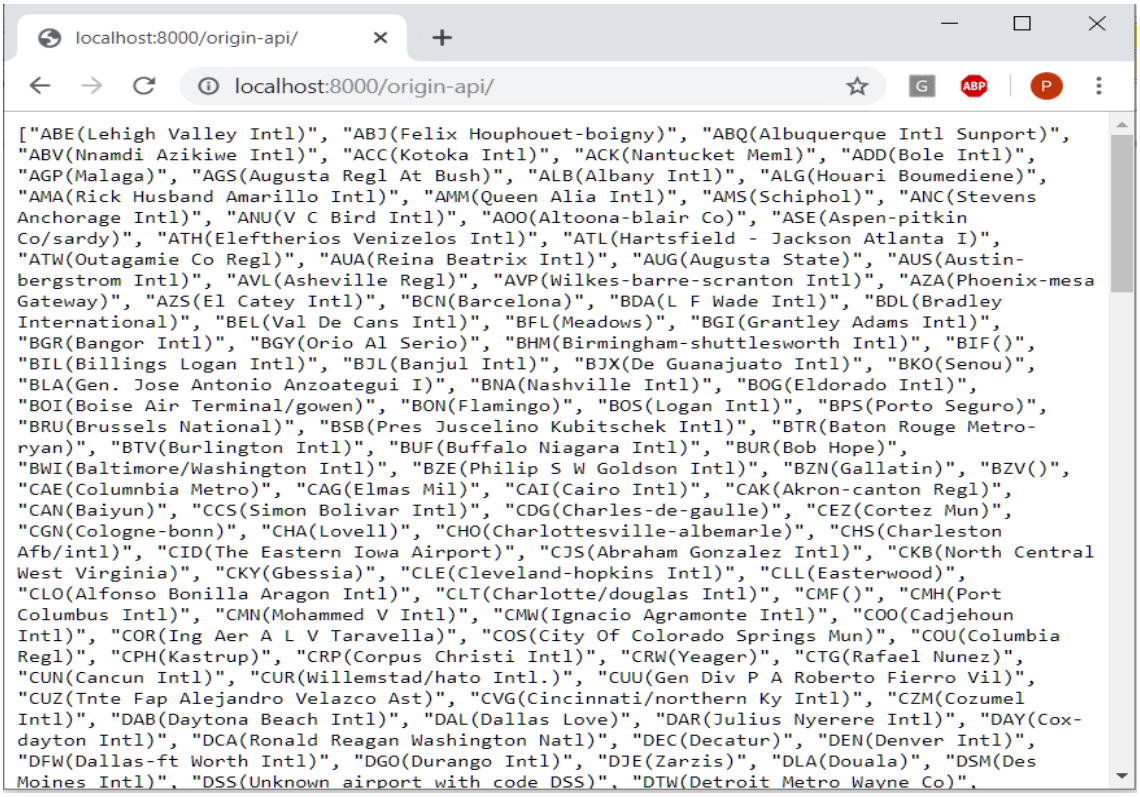
Implementation

REST API

Following REST APIs have been implemented to support this implementation:

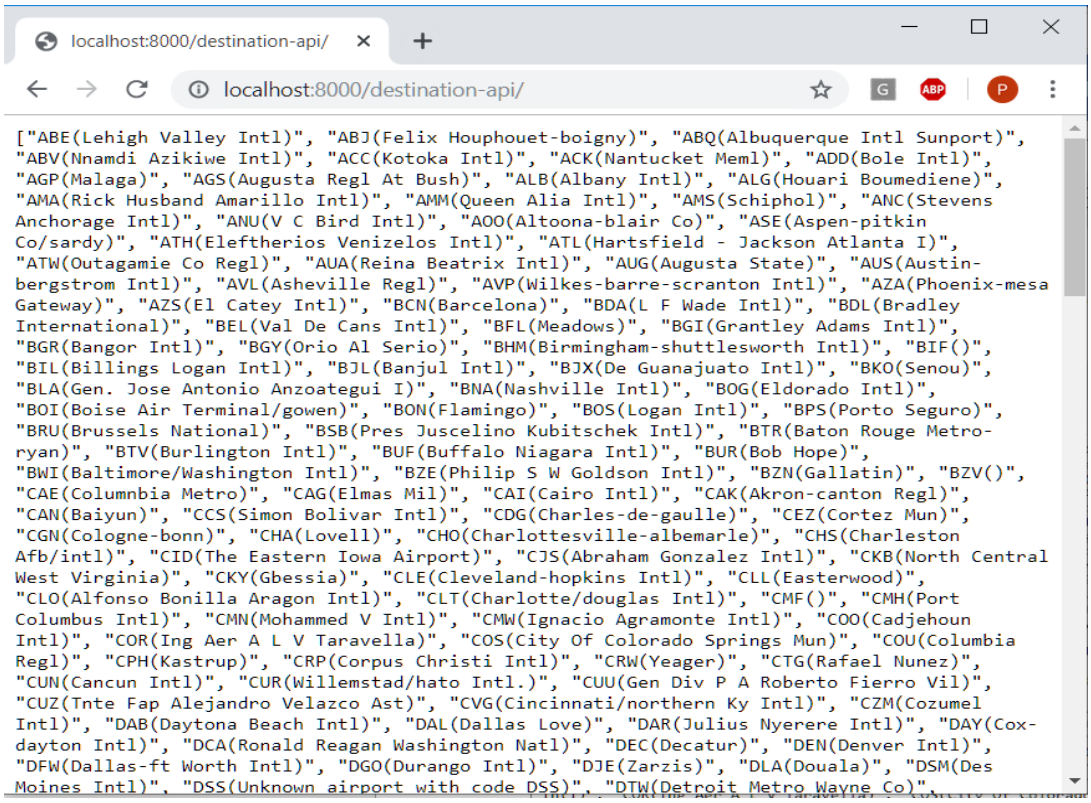
Origin-api

JSON response containing a unique list of all the Origins in the file. Consumed by the web application for the Auto-suggest feature



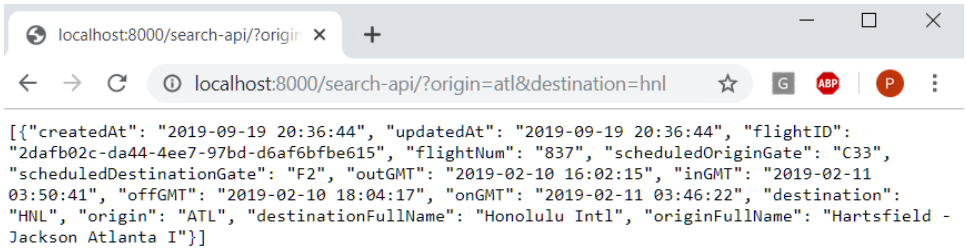
Destination-api

JSON response containing a unique list of all the destinations in the file. Consumed by the web application for the Auto-suggest feature



Search-api

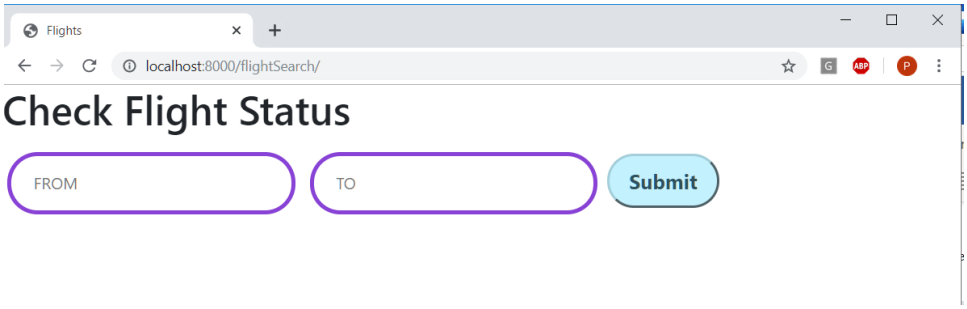
JSON response containing the flights returned from the search based on the origin and the destination that was passed



Web Application

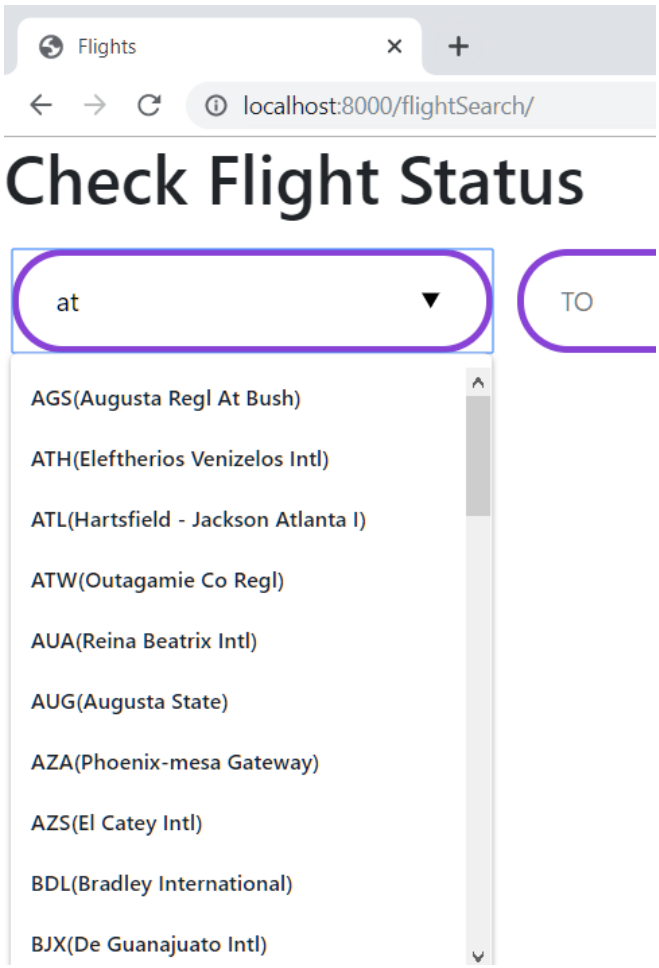
DJango based implementation to get a list of flights matching the search criteria. Search can be made using the origin or the destination or both.

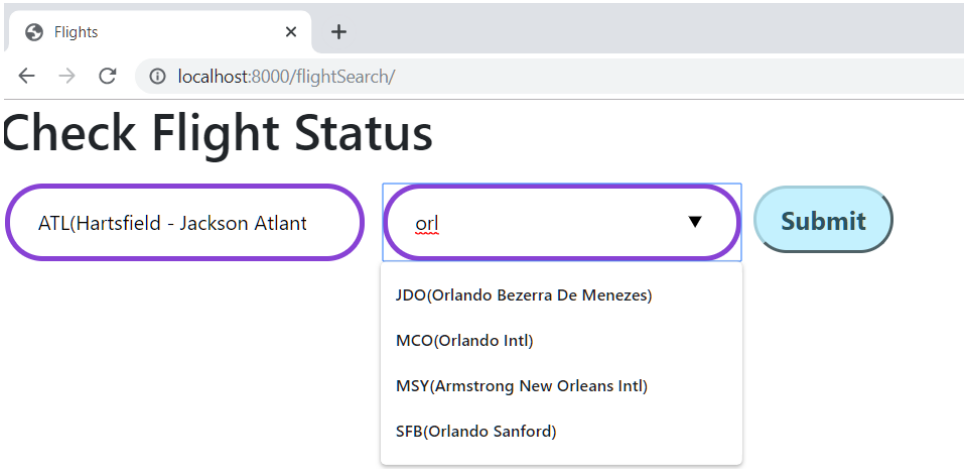
localhost:8000/flightSearch/



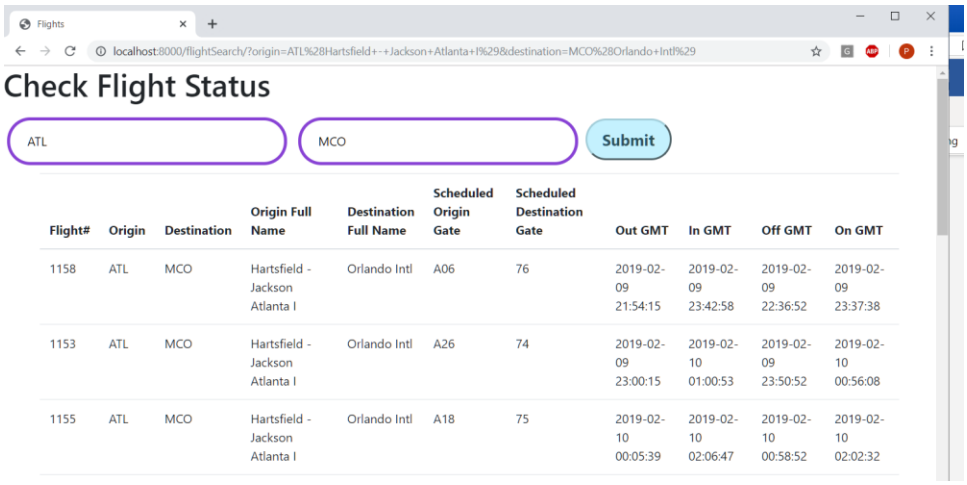
Auto Suggest

Auto suggest feature automatically suggests the available options for origin and destination as the user types on these fields. Suggestion is based on the values on both the airport code and the full name. Any substring matching the characters typed by the user is available in the suggestion. For example, typing ‘at’ in the origin field would suggest any of the origins that contain the string ‘at’ in the origin airport code or the full name.





Search results



Design consideration

Following design decisions were made with this implementation:

- Python Django based implementation – Django provides a rich framework for developing web applications based on the model-template-view architecture
- Security – Application did not implementation any security, since the data is not sensitive. Data is sent to the server using 'GET' requests as well.
- Bootstrap CSS framework has been leveraged for the front end
- JQuery/Javascript is used for the auto suggest, form validation and pagination
- Both APIs and the web application are implemented under the same project for simplicity. Ideally, we would want the api to be implemented on its own

Known issues:

Currently the auto-suggest feature seems slow to pull the suggestions. Will look at ways to optimize this implementation

Source files:

Source files are laid out based on the Django framework

Models.py - Contains the model class used for this implementation

Views.py - Contains the implementation for all the views (for the api and the web application)

Urls.py - Contains the routing information for the api calls and the web application

Templates – All the html templates used in this implementation are in this folder

Static files – CSS and javascript files are in this folder